UDACITY

Return to Classroom

Explore US Bikeshare Data

REVIEW

CODE REVIEW 7

HISTORY

Meets Specifications

Congratulations on passing the project on your first go!

It's rare for students to perform this well in the project and meet all the specifications correctly, but your hard work is made it possible. I really liked how you calculated the most popular trip by combining the start and end stations into one column. Really clever thinking!

Here's what you were able to accomplish in this project submission:

✅ All code cells can be run without error.

✅ Appropriate data types (e.g. strings, floats) and data structures (e.g. lists, dictionaries) are chosen to carry out the required analysis tasks.

✅ Loops and conditional statements are used to process the data correctly.

✅ Packages are used to carry out advanced tasks.

✅ Functions are used to reduce repetitive code.

✅ Docstrings, comments, and variable names enable the readability of the code.

✅ Raw input is solicited and handled correctly to guide the interactive question-answering experience; no errors are thrown when unexpected input is entered.

✅ Descriptive statistics are correctly computed and used to answer the questions posed about the data

✅ Raw data is displayed upon request by the user in an interactive manner in batches of 5 rows.

That's quite an accomplishment! You show a clear understanding of using Python to perform exploratory data analysis, and bundle this in a interactive program that allows its users to explore these statistics. This is no small feat and you should be really proud of yourself! 🚀

I have not much to add to this excellent submission. I have still left notes for improving the user experience, and improving your coding habits in this review. Please go through them in your own time.

I wish you the best of luck for the rest of the Nanodegree! Happy learning! 📚

Code Quality

All code cells can be run without error.

Tips: Implement safeguards against invalid user inputs that can potentially break the codes. Please refer to the “Solicit and handle raw user input” rubric item for further details.

Requirement Passed! ✅

Everything runs free of errors. Nicely done! 🌟

Suggestion 💡

Debugging i.e. dealing with errors, finding bugs and implementing fixes is an integral part of the programming. I believe you must have done a lot of it while working on this project. There are multiple ways to make the life of a software programmer easier. Here are some of them:

1. **Use an IDE with debugging capabilities** (like PyCharm) and use breakpoints to debug your code. You can download a free, community version of PyCharm from here. The official documentation on using PyCharm for debugging Python code can be found here.
2. **Use Visual Studio Code** as an alternative to above if you prefer that as your code editor of choice. Here is a great article to get started with that.
3. **Use a Python notebook** to write exploratory/prototype code and then transfer or convert it to a script. Python programmers and even scientists widely use "notebooks" as their programming environment of choice. The beauty of using a notebook is that you can write your code in notebook "cells" and immediately run it and see its output in the next cell. You can try Jupyter notebook, one of the most popular python notebook packages, here right in your browser! If it feels interesting, feel free to install it to your local machine and add it to your python toolbox.

Appropriate data types (e.g. strings, floats) and data structures (e.g. lists, dictionaries) are chosen to carry out the required analysis tasks.

Requirement Passed! ✅

You have appropriately used lists and dataframes to carry out the analysis tasks. Reading everything into a Pandas dataframe is one of the best first steps in making your data analysis project a success. You have done well here. 🌟

Further Reading 📖

You should have a good graps of the basic data types and structures in Python by going through this course. If you want to improve your knowledge of data structures even further, check out this resource. Data structures are fundamental to computer science, and practicing implementation of these will make you a better programmer.

Loops and conditional statements are used to process the data correctly.

Requirement Passed! ✅

Loops and conditionals are appropriately used to process user input and calculate the statistics. 🌟

Further Reading 📖

You'll be using loops and conditionals a lot when writing code, so it's very helpful to practice them from time to time.. If you want to practice your loops and conditionals, check out this resource. If you're interested in taking your programming skills up a few more notches, the Advent of Code exercises are excellent to help you do so!

Packages are used to carry out advanced tasks.

Requirement Passed! ✅

Pandas is a great choice for solving this project! Great work on implementing its functionality to work out the statistics! 🌟

Further Reading 📖

Python for Data Science: This is a great resource for a quick and practical guide to performing the most common data munging and analysis tasks with Pandas. Be sure to check it out!

Functions are used to reduce repetitive code.

Requirement Passed! ✅

You're appropriately used functions to organise the user input handling, data loading and filtering, and calculation of different statistics into separate chunks of code. This helps both the author, and readers of the code focus their attention of isolated/specific parts of code at a time. Great job with this!

Further Reading 📖

The next step for organising your code and writing more complex software is using Classes and Objects. Though using classes was outside the scope of this project, you're likely to use classes and objects if you choose to (or already) have a career relying on software engineering. This tutorial serves as an excellent introduction to the concepts, so be sure to check it out!

Docstrings, comments, and variable names enable the readability of the code.

Tips: Please refer to the Python's documentation PEP 257 -- Docstring Conventions. Example of docstring conventions:

```
def function(a, b):
    """Do X and return a list."""
```

Requirement Passed! ✅

You have appropriately used docstrings and comments, and your variable names help understand the kind of data being stored in that variable. 🌟

Further Reading 📖

Improving your coding style is something you will continue to do during the bulk of your programming career. For a comprehensive guide on good pythonic coding style, check out this resource: https://docs.python-guide.org/writing/style/

Script and Questions

Raw input is solicited and handled correctly to guide the interactive question-answering experience; no errors are thrown when unexpected input is entered.

User inputs should be made **case insensitive**, which means the input should accept the string of "Chicago" and its case variants, such as "chicago", "CHICAGO", or "cHicAgo".

You should also implement **error handlings** so your program does not throw any errors due to invalid inputs. For example, if the user enters "Los Angeles" for the city, the error handling should reject the user input and avoid breaking the codes.

Requirement Passed! ✅

Raw input is correctly handled including case-sensitivity. No errors are thrown on unexpected inputs. 🌟

Descriptive statistics are **correctly computed** and used to answer the questions posed about the data.

Raw data is displayed upon request by the user in the following manner:

- Your script should prompt the user if they want to see 5 lines of raw data,
- Display that data if the answer is 'yes',
- Continue iterating these prompts and displaying the **next** 5 lines of raw data at each iteration,
- Stop the program when the user says 'no' or there is no more raw data to display.

Tips: you can implement the `while` loop and track the row index in order to display the continuous raw data.

Requirement Passed! ✅

Statistics are calculated and displayed. Users are able to inspect the raw data in batches of 5 rows. 🌟

DOWNLOAD PROJECT

7 CODE REVIEW COMMENTS ›

RETURN TO PATH

Rate this review

START